THAT WHICH IS CLAIMED

1. A logic network for directly computing non-iteratively a quotient of a divisor and a dividend, said logic network comprising at least one stage having logic devices, wherein said logic devices subtract the divisor from the dividend and compare the divisor and dividend, wherein if the dividend is at least as great as the divisor, said logic devices set a most significant bit of the quotient to a one and output the difference between the divisor and dividend and elsewise set the most significant bit of the quotient to a zero and output the dividend, and wherein said logic devices of said stage perform all processing independent of a clock signal.

2. A logic network according to Claim 1, wherein said stage comprises:
at least one subtractor for subtracting the divisor from the dividend; and
a selector for setting a bit of the quotient and selecting between the dividend and the difference between the divisor and dividend for output.

3. A logic network according to Claim 1 comprising a plurality of stages, wherein a first stage generates the most significant bit of the quotient and each successive stage generates the next lesser significant bit of the quotient, and wherein said logic network between each stage includes a shifter that shifts the bits of the output from the previous stage one place to the left to thereby multiply the output by two.

4. A logic network according to Claim 3, wherein the quotient has h-bit accuracy for a dividend and a divisor, wherein h is a preselected positive integer >2, wherein said logic network includes h number of stages.

5. A logic network according to Claim 1 wherein the logic network is derived by universal approximators.

6. A method of directly computing a quotient of a divisor and a dividend in a non-iterative and unclocked manner comprising the steps of:

providing a logic network comprising logical devices for determining the quotient of the divisor and the dividend independent of a clock;

applying the divisor to a divisor input of the logic network;

applying the dividend to a dividend input of the logic network;

computing the quotient with the logic network non-iteratively and in an unclocked manner; and

acquiring the quotient from the logic network.

7. A method according to Claim 6, wherein said computing step computes each bit of the quotient in computing stages, wherein for at least one computing stage said method comprises the steps of:

subtracting the divisor from the dividend;

comparing the divisor and dividend;

setting a bit of the quotient to a one if the dividend is at least as great as the divisor and elsewise setting the bit of the quotient to a zero; and

outputting the difference between the divisor and dividend if the dividend is at least as great as the divisor and elsewise outputting the dividend.

8. A method according to Claim 7, wherein said computing step computes each bit of the quotient in a plurality of computing stages, and wherein between each computing stage said method further comprises the step of left shifting the bits of the output from the previous stage one place to thereby multiply the output by two.

9. A method according to Claim 6 further comprising the step of generating the logic network using universal approximators.

10. A method of computing a quotient having h-bit accuracy for a dividend and a divisor, wherein h is a preselected positive integer >2, the method being non-iterative and performing the following process in each stage of a non-iterative logic network, wherein the first stage generates a most significant bit of the quotient based on the dividend and divisor and each successive stage generates the next lesser significant bit based on an incoming dividend output from the previous stage and the divisor, the method comprising the steps of:

1) making an underflow true if the divisor is greater than the incoming dividend, and elsewise making the underflow false if the divisor is not greater than the incoming dividend;
2) making the output dividend twice the incoming dividend if underflow is true, and elsewise making the output dividend twice the difference between and the divisor and the incoming dividend if underflow is false;
3) making the current stage quotient output bit 0 if underflow is true, and elsewise making the current stage quotient output bit 1 if underflow is false;
4) repeating steps 1), 2), and 3) with each stage after the first stage, receiving the output dividend of its preceding stage and the divisor, until h-1 bits of the quotient are determined, then in the final stage;
5) making the underflow true if the divisor is greater than the incoming dividend, and elsewise making the underflow false if the divisor is not greater than the incoming dividend; and
6) making the current stage quotient output bit 0 if underflow is true, and elsewise making the current stage quotient output bit 1 if underflow is false.

11. A method according to Claim 10, wherein the radix is other than two.

12. A method according to Claim 10, wherein a part of at least one stage of said method is replaced by a precomputed table.

13. A logic network for computing the reciprocal of a number M in a non-iterative manner comprising logical devices configured to calculate an algebraic function that is a relationship of at least two numbers X and A that sum to equal the number M, wherein said logic network operates independent of a clock signal.

14. A logic network according to Claim 13, wherein said logic network has logical devices configured to calculate the algebraic function:

$1/M = 1/X - A/(X^2 + AX)$.

15. A logic network according to Claim 13, wherein said logic network has logical devices configured to calculate the algebraic function that approximates the reciprocal of number M to a predetermined accuracy, said algebraic function being:

$1/M \approx (X - A)/X^2$ for $A < X$.

16. A logic network according to Claim 13, wherein said logic network is derived by universal approximators.

17. A logic network according to Claim 13, wherein said logic network includes storage devices for storing all possible values for at least some portions of the algebraic function, such that the stored values can be addressed and used in computing the reciprocal of the number M.

18. A method of computing a reciprocal of a number M, comprising the steps of:

separating number M into at least two numbers X and A so that number M equals the sum of the at least two numbers X and A;

generating an algebraic function relating the at least two numbers X and A to the reciprocal of number M; and

computing the reciprocal of number M according to the algebraic function.

19. A method according to Claim 18, wherein said generating step generates an equation that equates 1/M to an algebraic function of the at least two numbers.

20. A method according to Claim 19, wherein said generating step generates the algebraic function of:

$1/M = 1/X - A/(X^2 + AX)$.

21. A method according to Claim 18, wherein said generating step generates the algebraic function that approximates the reciprocal of number M to a predetermined accuracy.

22. A method according to Claim 21, wherein said generating step generates the algebraic function based on the requirement that one of the at least two numbers X and A, is much smaller than the other of the at least two numbers.

23. A method according to Claim 22, wherein said generating step generates the following algebraic function that approximates the reciprocal of number M:

$1/M \approx (X - A)/X^2$ for $A < X$.

24. A method according to Claim 18, wherein said generating step generates the algebraic function relating the at least two numbers X and A to the reciprocal of the number M, wherein the algebraic function includes a reference portion that is the inverse of a part of number M and a correction portion for correcting the reference portion.

25. A method according to Claim 24, wherein said generating step generates the algebraic function relating the at least two numbers X and A to the reciprocal of the number M, wherein the algebraic function includes a reference portion that is the inverse of a part of number M and a correction portion subtracted from the reference portion.

26. A method according to Claim 24, wherein said generating step generates the algebraic function relating the at least two numbers X and A to the reciprocal of the number M, wherein the algebraic function includes a reference portion that is the inverse of a part of number M and a correction portion multiplied to the reference portion.

27. A method according to Claim 18 further comprising the steps of calculating and storing all possible values for at least some portions of the algebraic function, and wherein said computing step comprises accessing the store values to compute the reciprocal of the number M.

28. A method of computing the reciprocal of a number M in a non-iterative manner comprising the steps of:

providing a logic network comprising logical devices for determining the reciprocal of number M, wherein the logical devices are configured to calculate an algebraic function that is a relationship of at least two numbers X and A that

sum to equal the number M, wherein said logic network operates independent of a clock signal;

applying the number M to the logic network; and

acquiring the reciprocal of the number M from an output of the logic network.

29. A method according to Claim 28, wherein said providing step provides a logic network having logical devices configured to calculate the algebraic function:

$1/M = 1/X - A/(X^2 + AX)$.

30. A method according to Claim 28, wherein said providing step provides a logic network having logical devices configured to calculate an algebraic function that approximates the reciprocal of number M to a predetermined accuracy, said algebraic function being:

$1/M = 1/X - A/(X^2 + AX)$

simplified using the approximation

$1/(X+A) \approx (X-A)/X^2$ for $A < X$.

31. A method of computing the reciprocal of a number M in a non-iterative manner comprising the steps of:

providing a logic network that operates independent of a clock signal and comprising logical devices for determining the reciprocal of number M, wherein the logical devices are configured to calculate an algebraic function that is a relationship of at least two numbers X and A that sum to equal the number M, said algebraic function approximating the reciprocal of number M to a predetermined accuracy and being:

$1/M = 1/X - A/(X^2 + AX)$

simplified using the approximation

$1/(X+A) \approx (X-A)/X^2$ for $A < X$;

applying the number M to the logic network; and

acquiring the reciprocal of the number M from an output of the logic network.

32. A method according to Claim 31 further comprising the step of multiplying the number M as a divisor to a dividend to determine a quotient.

33. A computer program product for computing a reciprocal of a number M, wherein the computer program product comprises:

a computer readable storage medium having computer readable program code means embodied in said medium, said computer-readable program code means comprising:

first computer instruction means for separating number M into at least two numbers X and A so that number M equals the sum of the at least two numbers X and A;

second computer instruction means for generating an algebraic function relating the at least two numbers X and A to the reciprocal of number M; and

third computer instruction means for computing the reciprocal of number M according to the algebraic function.

34. A computer program product according to Claim 33, wherein said second computer instruction means generates an equation that equates 1/M to the algebraic function of the at least two numbers.

35. A computer program product according to Claim 33, wherein said second computer instruction means generates the algebraic function of:

$1/M = 1/X - A/(X^2 + AX)$.

36. A computer program product according to Claim 33, wherein said second computer instruction means generates the algebraic function that approximates the reciprocal of number M to a predetermined accuracy.

37. A computer program product according to Claim 36, wherein said second computer instruction means generates the algebraic function based on the requirement that one of the at least two numbers X and A, is much smaller than the other of the at least two numbers.

38. A computer program product according to Claim 37, wherein said second computer instruction means generates the following algebraic function that approximates the reciprocal of number M, said algebraic function being:

$1/M = 1/X - A/(X^2 + AX)$

simplified using the approximation

$1/(X+A) \approx (X-A)/X^2$ for $A < X$.